



Javascript avancé

Module 1 - Rappels



OBJECTIFS



Javascript avancé
OBJECTIFS

Apprendre les bases du
javascript

Savoir manipuler le DOM

Tester du code javascript



SOMMAIRE



Introduction



Type de données

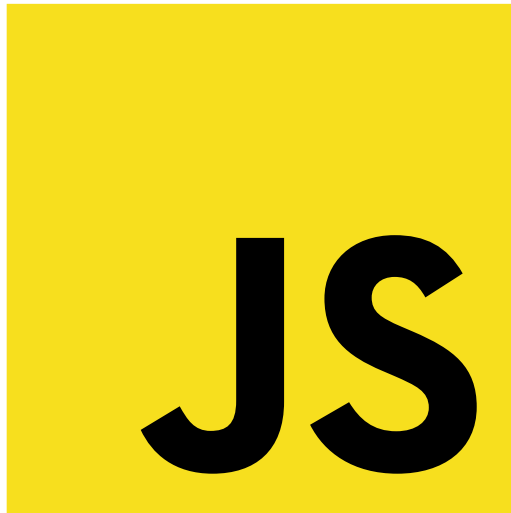


Structure de code



DOM

Introduction

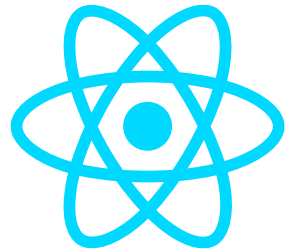


- Créé en 10 jours par Brendan Eich en 1995
- Inspiré de Java
- Exécuté coté client
- Full stack
- Orienté objet
- Multi paradigme

Javascript fait parti du trio
qui a fait la gloire du navigateur Netscape



Les différents frameworks Javascript



2009 – ES5

2015 – ES6

2016 – ES7

2017 – ES8

ES6 est la plus importante mise à jour du langage

- Une fonctionnalité n'est jamais retirée
- Les mises à jour sont incrémentales
- Un site web qui fonctionnait fonctionnera toujours

Types de données

Type de données

- string
- boolean
- number
- undefined
- null
- bigint (es6)

```
console.group('Types de données');

let prenom = 'Chloe';
let estAdmin = true;
let age = 29;

let chloe = {
  prenom: 'Chloe',
  estAdmin: true,
  age: 29
}

console.log(chloe);

console.groupEnd();
```

Tableaux



```
let villes = [  
  'Brest',  
  'Quimper',  
  'Nantes',  
  'Rennes'  
];
```

Un tableau est un objet



```
console.log(villes.length); // 4  
console.log(villes.sort()); // ['Brest', 'Nantes', 'Quimper', 'Rennes']
```

Structures de code

Structure de code

Conditions :

```
const nombre = 42;

if (nombre === 42) {
  console.log("Quelle était la question ?");
} else {
  console.log("Ce n'est pas la bonne réponse.");
}
```

Boucles

```
for (let i = 0; i < 29; i++) {
  // Ternaire
  console.log(
    i % 5 ?
    `${i} est un multiple de 5` :
    `${i} n'est pas un multiple de 5`);
}
```

Fonctions et portée

Fonctions



```
/**
 * @param nombre1
 * @param nombre2
 * @returns number
 */
function addition(nombre1, nombre2) {
  return nombre1 + nombre2;
}
```



```
/**
 * !\ On ne tient pas compte du mois de naissance !\ 🤖
 * @param anneeDeNaissance
 * @returns number
 */
const nbAnneesDepuis = function (anneeDeNaissance) {
  return new Date().getFullYear() - anneeDeNaissance;
}
console.log(nbAnneesDepuis(1984));
```



```
// Fonction fléchée 🤖
const nbAnneesFlecheesDepuis =
  anneeDeNaissance => new Date().getFullYear() - anneeDeNaissance;
console.log(nbAnneesFlecheesDepuis(1984));
```

Portée

- La portée globale :
- La portée de fonctions :
- La portée de bloc :

```
const ecole = 'ENI école';
```

```
function addition(nombre1, nombre2) {  
  const somme = nombre1 + nombre2;  
  return somme;  
}  
  
console.log(somme); //  ERREUR
```

```
if (41 < 42) {  
  const resultat = true;  
}  
  
console.log(resultat); //  ERREUR
```


DÉMONSTRATION

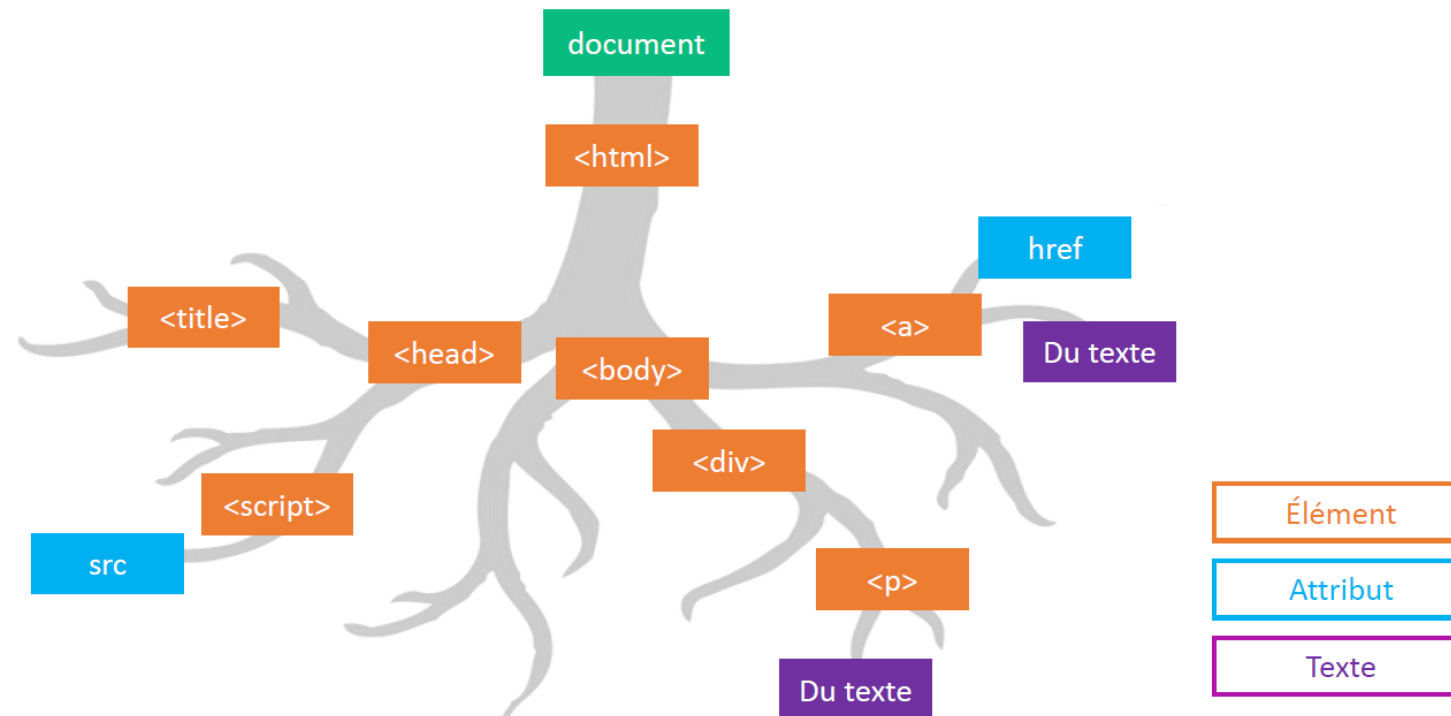
Données, structure de code et fonctions



Document Object Model

Document Object Model

Le DOM est la représentation d'un document HTML



Manipulation du DOM

Javascript permet l'accès et la manipulation de chaque élément de l'arbre.

- `document.getElementById("");`
- `document.querySelector("");`

- `monElement.innerText = "";`
- `monElement.src = "";`




getElement()



```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>DOM</title>
  <script src="script.js"></script>
</head>
<body>
<p id="ecole" class="ecolecss">ENI école</p>
<script>
  const ecole1 = document.getElementById('ecole');
  const ecole2 = document.querySelector('.ecolecss');
  console.log(ecole1 === ecole2); // true 👍
</script>
</body>
</html>
```

Evènements

Javascript permet également l'utilisation d'évènements

-  click, dbclick, mouseenter, mousemove, wheel
-  keydown, keypress, keyup
-  select, change, blur, focus

On <Evenement>

```
● ● ●

<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>DOM</title>
</head>
<body>
<p id="ecole">ENI école</p>
<script>
  document.getElementById("ecole").addEventListener("click", clic);
  function clic() {
    console.log("J'ai cliqué");
  }
</script>
</body>
</html>
```

DÉMONSTRATION

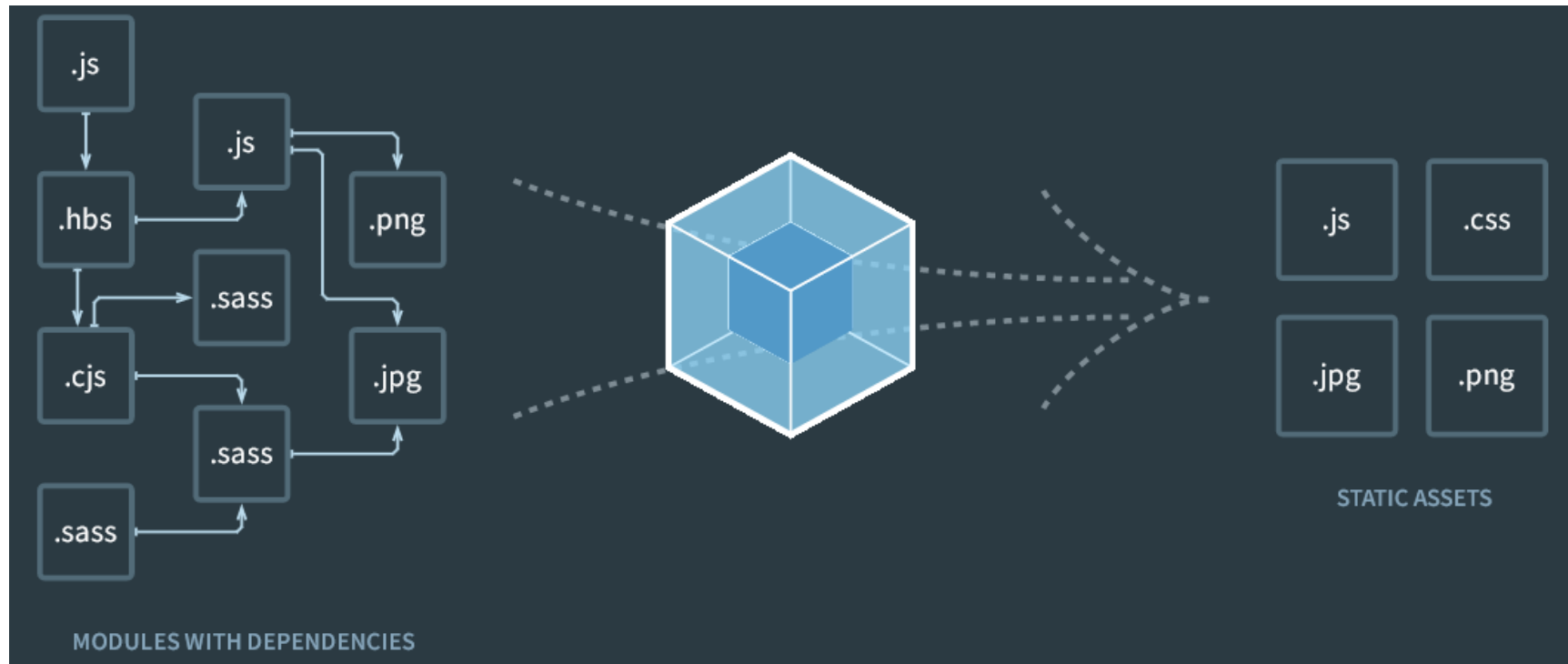
Le DOM



Bundler

Webpack

Webpack permet, à partir de plusieurs fichiers (html, css, javascript) de fournir un produit fini, minifié et fonctionnel,



DÉMONSTRATION

Webpack



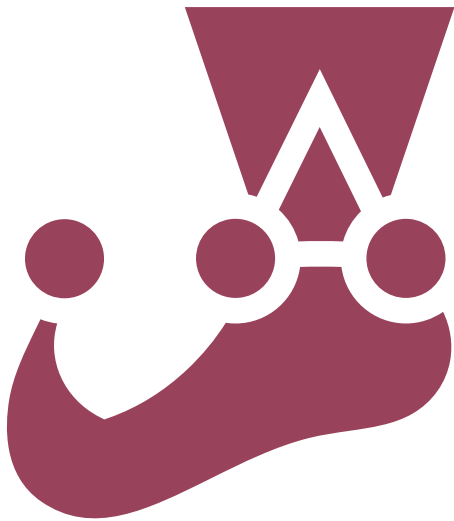
TRAVAUX PRATIQUES

Devinette



Tests

JEST



```
describe('scenario de test', () => {  
  it('devrait marcher', () => {  
    expect(true).toBe(true);  
  });  
});
```

JEST est un framework de tests mettant l'accent sur la simplicité

DÉMONSTRATION

Jest



Conclusion

- Vous connaissez les bases du Javascript
- Vous savez manipulez le DOM grâce au Javascript
- Vous savez rédiger des tests en Javascript